CLAIMS

We claim:

SuB B17

1. A method of restructuring a program comprising basic blocks ~~for execution~~ by a processor having a memory hierarchy comprising a plurality of levels, said method comprising the steps of:

a) constructing a Program Execution Graph (PEG) from control flow and frequency information from a profile of the program, the PEG comprising a weighted undirected graph comprising nodes representing the basic blocks and edges representing transfer of control between pairs of the basic blocks, each of the nodes having a weight equal to the size of the basic block represented by the node, each of the edges having a weight equal to a frequency of transition between a pair of basic blocks represented by a pair of nodes connected by the edge;

b) partitioning the nodes of the PEG into clusters such that a sum of weights of the edges whose endpoints are in different clusters is minimized, and such that for any cluster, a sum of weights of the nodes in the cluster is no greater than an upper bound; and

c) restructuring the basic blocks into contiguous code corresponding to the clusters.

2. The method of claim 1 further comprising the steps of:

d) constructing a next PEG from the clusters of the partitioned PEG such that a node in the next PEG corresponds to a cluster in the partitioned PEG, and such that there is an edge between two nodes in the next PEG if there is an edge between components of the clusters represented by the two nodes; and

e) assigning a weight to each node of the next PEG; and

f) assigning a weight to an edge between a pair of nodes of the next PEG representing a pair of clusters of the partitioned PEG, the edge weight being a summation of weights of edges in the partitioned PEG having endpoints in the pair of clusters in the partitioned PEG.

3. The method of claim 2 further comprising the step of:

f) repeating steps b through f.

4. The method of claim 1 wherein the upper bound is a multiple of a size of a level of the memory hierarchy.

Sub B2

5. ~~The method of claim 3 wherein the upper bound for a level of~~ the memory hierarchy, other than a first level, is a ~~size of the~~ memory hierarchy level divided by an upper bound used to partition a next lower level of the memory ~~hierarchy.~~

6. The method of claim 3 further comprising the steps of:

removing a basic block whose size is greater than the upper bound from the partitioning step; and

reintegrating the basic block whose size is greater than the upper bound into a next repetition of steps b through f.

7. An article of manufacture for use in a computer system for restructuring a program comprising basic blocks for execution by a processor having a memory hierarchy comprising a plurality of levels, said article of manufacture comprising a computer-readable storage medium having a computer program embodied in said medium which may cause the computer system to:

a) construct a Program Execution Graph (PEG) from control flow and frequency information from a profile of the program, the PEG comprising a weighted undirected graph comprising nodes representing the basic blocks and edges representing transfer of control between pairs of the basic blocks, each of the nodes having a weight equal to the size of the basic block represented by the node, each of the edges having a weight equal to a frequency of transition between a pair of basic blocks represented by a pair of nodes connected by the edge;

b) partition the nodes of the PEG into clusters such that a sum of weights of the edges whose endpoints are in different clusters is minimized, and such that for any cluster, a sum of weights of the nodes in the cluster is no greater than an upper bound; and

c) restructure the basic blocks into contiguous code corresponding to the clusters.

8. The article of manufacture of claim 7 wherein the computer program may further cause the computer system to:

d) construct a next PEG from the clusters of the partitioned PEG such that a node in the next PEG corresponds to a cluster in the partitioned PEG, and such that there is an edge between two nodes in the next PEG if there is an edge between components of the clusters represented by the two nodes; and

e) assign a weight to each node of the next PEG; and

f) assign a weight to an edge between a pair of nodes of the next PEG representing a pair of clusters of the partitioned PEG, the edge weight being a summation of weights of edges in the partitioned PEG having endpoints in the pair of clusters in the partitioned PEG.

9. The article of manufacture of claim 8 wherein the computer program may further cause the computer system to:

c f) ~~repeat steps~~ b through f.

10. The article of manufacture of claim 7 wherein the upper bound is a multiple of a size of a level of the memory hierarchy.

~~11.~~ ~~The article of manufacture of claim 10 wherein the upper bound for a level of the memory hierarchy, other than a first level,~~ is a size of the memory hierarchy level divided by an upper bound used to partition a next lower level of the memory ~~hierarchy~~.

12. The article of manufacture of claim 9 wherein the computer program may further cause the computer system to:

remove a basic block whose size is greater than the upper bound from the partitioning step; and

reintegrate the basic block whose size is greater than the upper bound into a next repetition of steps b through f.

13. A computer system for restructuring a program comprising basic blocks for execution by a processor having a memory hierarchy comprising a plurality of levels, said computer system comprising:

a) a Program Execution Graph (PEG) constructed from control flow and frequency information from a profile of the program, the PEG comprising a weighted undirected graph comprising nodes representing the basic blocks and edges representing transfer of control between pairs of the basic blocks, each of the nodes having a weight equal to the size of the basic block represented by the node, each of the edges having a weight equal to a frequency of transition between a pair of basic blocks represented by a pair of nodes connected by the edge;

b) a partition of the nodes of the PEG into clusters such that a sum of weights of the edges whose endpoints are in different clusters is minimized, and such that for any cluster, a sum of weights of the nodes in the cluster is no greater than an upper bound; and

c) a restructuring of the basic blocks into contiguous code corresponding to the clusters.

14. The computer system of claim 13 further comprising:

d) a next PEG constructed from the clusters of the partitioned PEG such that a node in the next PEG corresponds to a cluster in the partitioned PEG, and such that there is an edge between two nodes in the next PEG if there is an edge between components of the clusters represented by the two nodes; and

e) a weight assigned to each node of the next PEG; and

f) a weight assigned to an edge between a pair of nodes of the next PEG representing a pair of clusters of the partitioned PEG, the edge weight being a summation of weights of edges in the partitioned PEG having endpoints in the pair of clusters in the partitioned PEG.

15. The computer system of claim 14 further comprising:

f) a repetition of elements b through f.

16. The computer system of claim 13 wherein the upper bound is a multiple of a size of a level of the memory hierarchy.

Sub B4

17. ~~The computer system of claim 15 wherein the upper bound~~ for a level of the memory hierarchy, other than a ~~first level,~~ is a size of the memory hierarchy level divided by an upper bound used to partition a next lower level of the memory ~~hierarchy~~.

18. The computer system of claim 15 further comprising:

a removal of a basic block whose size is greater than the upper bound from the partitioning step; and

a reintegration of the basic block whose size is greater than the upper bound into a next repetition of elements b through f.